SpookChain: Chaining a Sponge-Based AEAD with Beyond-Birthday Security

Gaëtan Cassiers, **Chun Guo**, Olivier Pereira, Thomas Peters, François-Xavier Standaert

SPACE 2019
06/12/2019

Our Contributions

- We present a new online Authenticated Encryption (AE) mode SpookChain
  - From the leakage-resilient AE mode of the NIST submission Spook
  - Leakage-resilience/built-in side-channel security
  - Provable online AE security, w.r.t. the dOAE model
  - Beyond n/2-bit “birthday-bound” security in the multi-user setting

Outline

- **Online Authenticated Encryption (AE)**
- Leakage-resilience
- Our contribution SpookChain mode

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
  - Ciphertext integrity INI-CTXT + Chosen-plaintext confidentiality IND-CPA (Bellare and Namprempre)

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
  - Ciphertext integrity INI-CTXT + Chosen-plaintext confidentiality IND-CPA (Bellare and Namprempre)
  - All-in-one definition: Rogaway and Shrimpton, EUROCRYPT 2006
    - Encryption always give rise to pseudorandom ciphertexts
    - Decryption always give rise to failure message $\perp$ / forgery trials never succeed

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
  - Ciphertext integrity INI-CTXT + Chosen-plaintext confidentiality IND-CPA (Bellare and Namprempre)
  - All-in-one definition: Rogaway and Shrimpton, EUROCRYPT 2006
  - De facto standard for real-world applications
    - CAESAR competition
    - NIST lightweight competition

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
- Online
  - Intuition: encryption need not to wait for all the plaintext blocks

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
- Online
  - Intuition: encryption need not to wait for all the plaintext blocks
  - Somewhat formal: encryption can be realized with constant memory cost [Hoang et al., CRYPTO 2015]

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
- Online
  - Intuition: encryption need not to wait for all the plaintext blocks
  - Somewhat formal: encryption can be realized with constant memory cost [Hoang et al., CRYPTO 2015]
  - Crucial for streaming processing: e.g., video transformation, upload and download huge files.

Online Authenticated Encryption (AE)

- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
- Online: Segmented-AE [Hoang et al., CRYPTO 2015]
  - View the long plaintext as a sequence of “segments” with sizes specified by the user
  - Segments are encrypted in turn
  - Encryption may not be online within a segment, but online between segments

■ Segmented-AE

A segmented-AE scheme is a triple $\Pi = (\boldsymbol{K}, \boldsymbol{E}, \boldsymbol{D})$ where the key space K is a non-empty set with an associated distribution and both encryption $\boldsymbol{E} = (\boldsymbol{E}.init, \boldsymbol{E}.next, \boldsymbol{E}.last)$ and decryption $\boldsymbol{D} = (\boldsymbol{D}.init, \boldsymbol{D}.next, \boldsymbol{D}.last)$ are specified by triple of deterministic algorithms such that

- $\boldsymbol{E}_K.init: \boldsymbol{N} \to \boldsymbol{S}$
- $\boldsymbol{E}_K.next: \boldsymbol{S} \times \boldsymbol{A} \times \boldsymbol{M} \to \boldsymbol{C} \times \boldsymbol{S}$
- $\boldsymbol{E}_K.last: \boldsymbol{S} \times \boldsymbol{A} \times \boldsymbol{M} \to \boldsymbol{C}$

$\boldsymbol{D}_K.init: \boldsymbol{N} \to \boldsymbol{S}$

$\boldsymbol{D}_K.next: \boldsymbol{S} \times \boldsymbol{A} \times \boldsymbol{C} \to \boldsymbol{M} \cup \{\perp\} \times \boldsymbol{S}$

$\boldsymbol{D}_K.last: \boldsymbol{S} \times \boldsymbol{A} \times \boldsymbol{C} \to \boldsymbol{M} \cup \{\perp\}$

where $K \leftarrow \boldsymbol{K}$, $\boldsymbol{N}$ is the nonce space, $\boldsymbol{S}$ is the state space, $\boldsymbol{A}$ is associated data space, $\boldsymbol{M}$ is the message space and $\boldsymbol{C}$ is the ciphertext space.

Online Authenticated Encryption (AE)

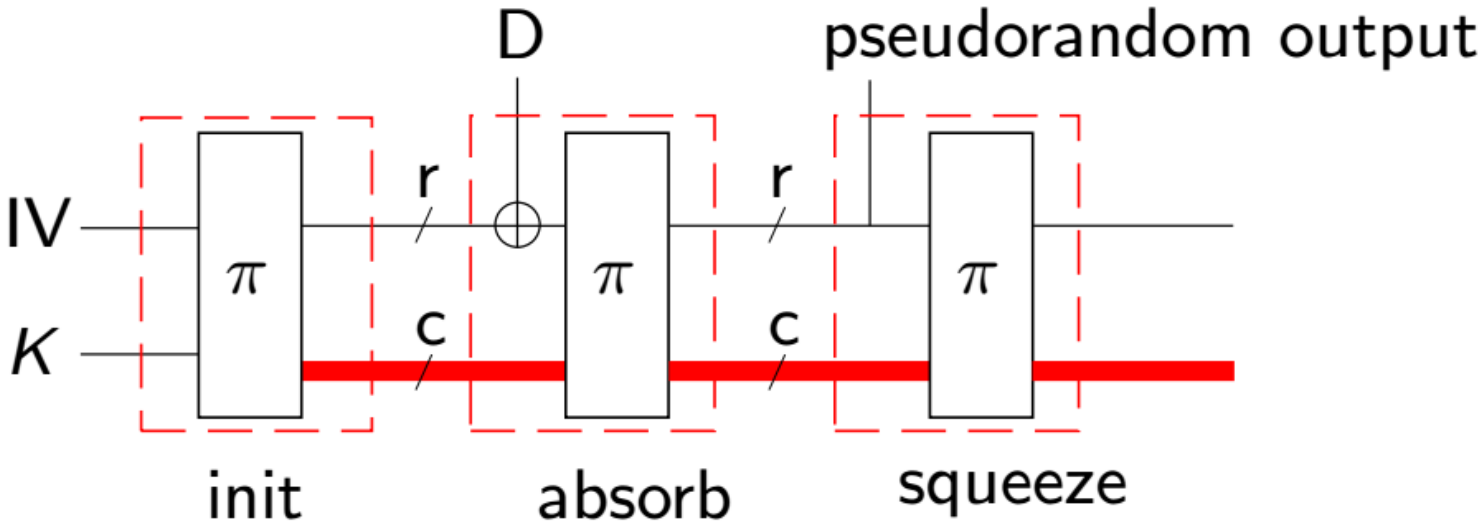
- AE: confidentiality and authenticity with a single primitive [Bellare and Namprempre, AISACRYPT 2000]
- Online: Segmented-AE [Hoang et al., CRYPTO 2015]
- Security
  - OAE: [Fleischmann et al., FSE 2012] (for “block-based” online AEs)
  - OAE2: best possible security for Segmented-AE [Hoang et al., CRYPTO 2015]
  - dOAE: somewhat best possible security for duplex-based online AEs [Bertoni et al., SAC 2011; Hoang et al., CRYPTO 2015]

Outline

- Online Authenticated Encryption (AE)
- **Leakage-resilience**
- Our contribution SpookChain mode

Leakage-resilient AEs

- Leakage-resilient AEs have built-in mechanisms to limit the damage of side-channel leakages
  - Typically achieved via consistently refreshing keys/states

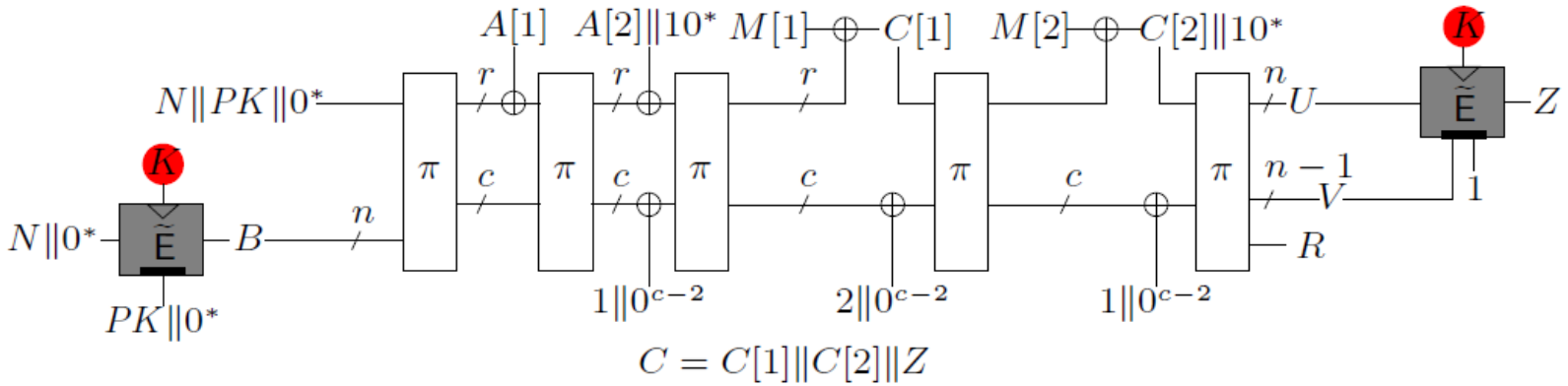


Leakage-resilient AEs

- Leakage-resilient AEs have built-in mechanisms to limit the damage of side-channel leakages
- Recent work on leakage-resilience of sponge/duplex, with different focuses and assumptions
  - Dobraunig and Mennink, Leakage Resilience of the Duplex Construction, eprint 2019/225.
  - Degabriele, Janson, and Struck, Sponges Resist Leakage: The Case of Authenticated Encryption, eprint 2019/1034.
  - Guo, Pereira, Peters, and Standaert, Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction, eprint 2019/193.

Leakage-resilient AEs

- Leakage-resilient AEs have built-in mechanisms to limit the damage of side-channel leakages
- Recent work on leakage-resilience of sponge/duplex, with different focuses and assumptions
- An example for illustration



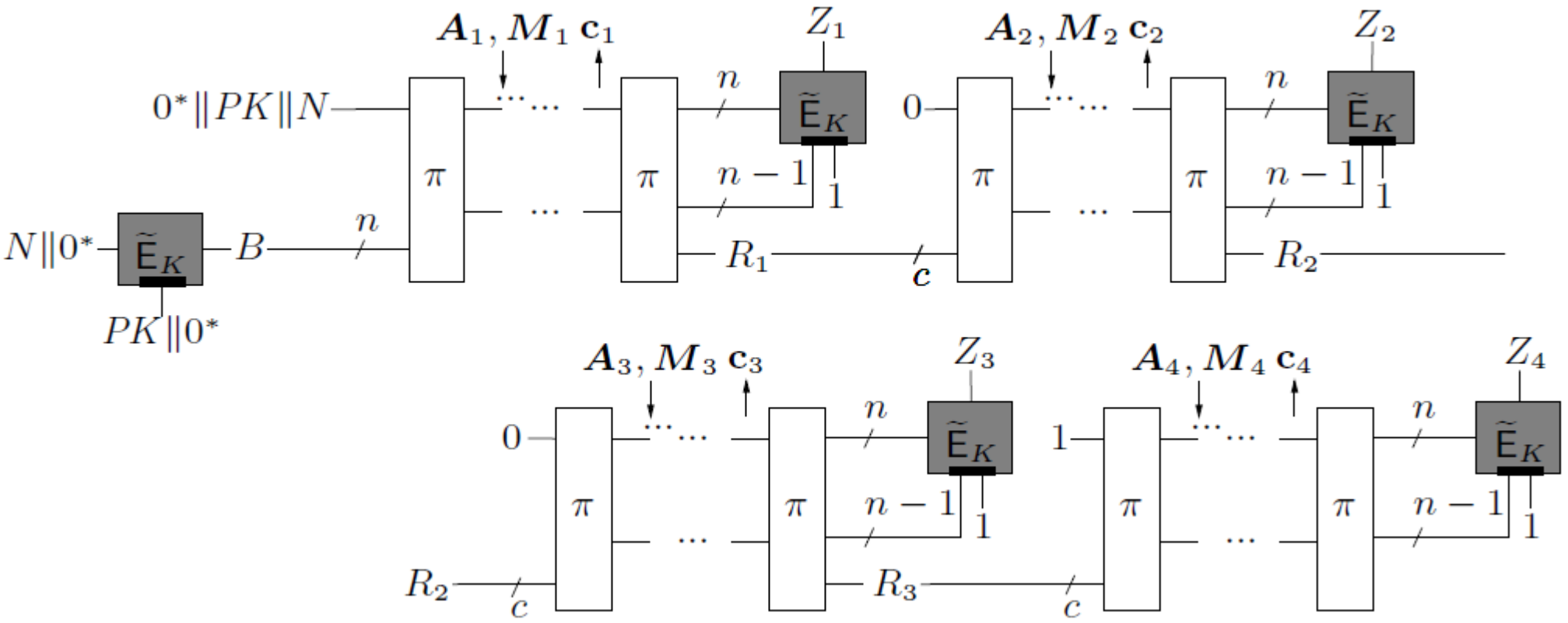
Outline

- Online Authenticated Encryption (AE)
- Leakage-resilience
- **Our contribution SpookChain mode**

Motivation

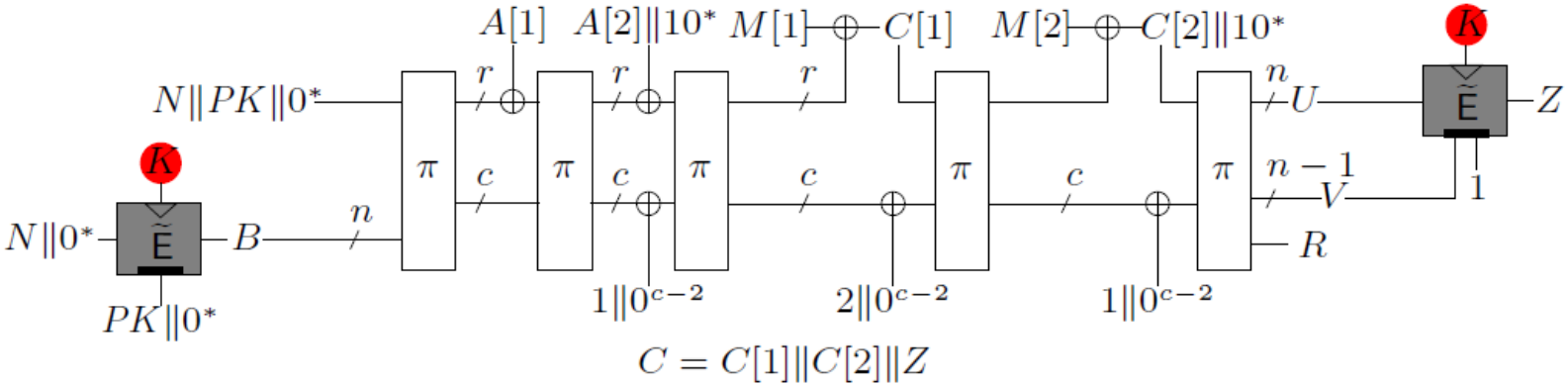
- Extending the mode (named TETSponge, eprint 2019/193) underlying the NIST lightweight AE submission Spook, to support **segment processing**
- The first attempt to combine **(provable) leakage-resilience** with **online computability**

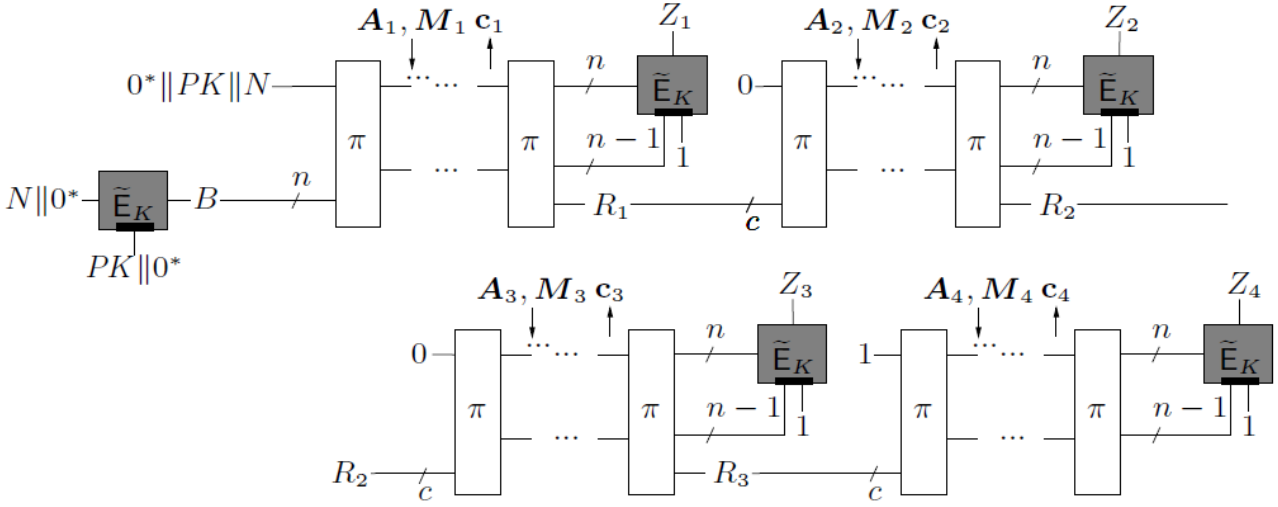
The mode in a nutshell



A formal description could be found in our paper

■ … comparison to TETSponge

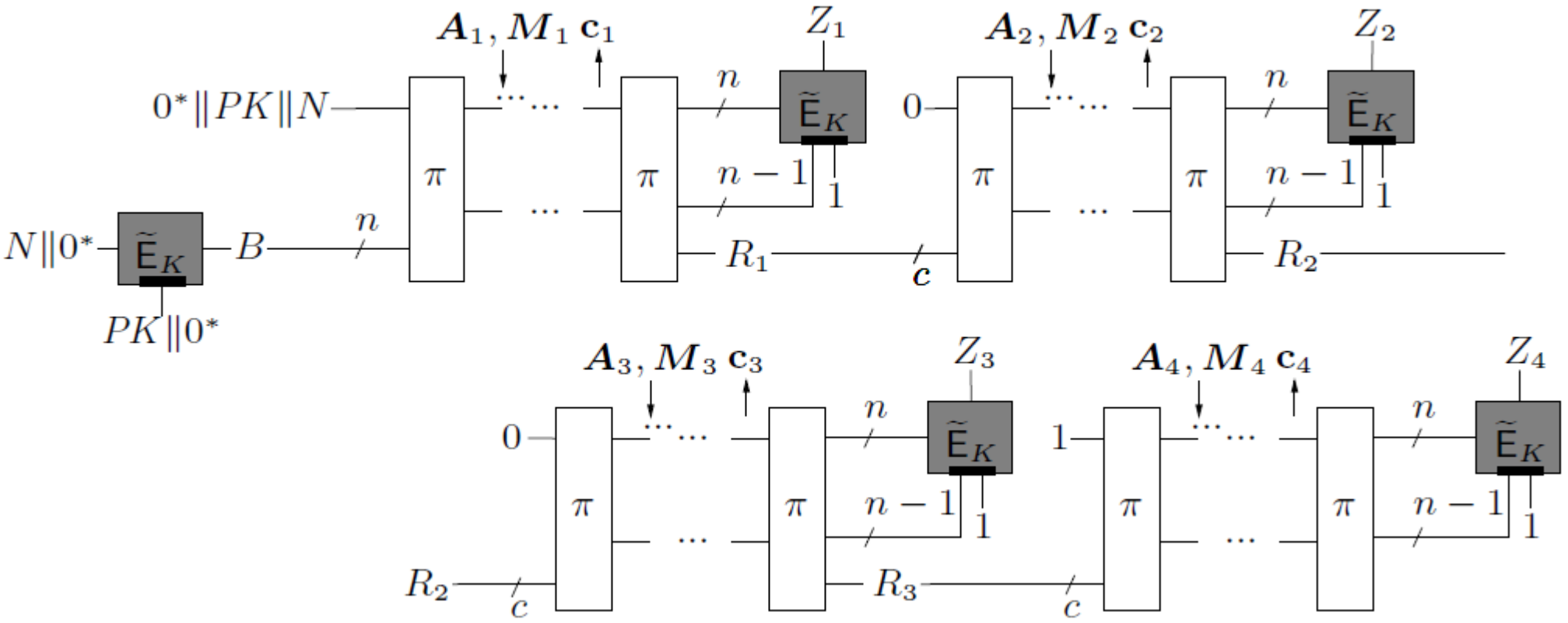




Security of SpookChain

- Duplex-based ⇒ **dOAE** security model
  - Introduced by Hoang et al. (CRYPTO 2015) to capture the best possible security achievable by duplex-based AEs
  - Intuition: starting from an already obtained plaintext-ciphertext pair, the adversary could not “append” a valid segment

■ What does dOAE security mean



Extending dOAE to the multi-user setting

- A natural extension: just include multiple users into the security game
- Motivation: an AE scheme will be deployed en mass and used by many computers/network sessions with many keys [Bellare(B) and Tackmann, CRYPTO 2016]
  - An adversary may find it satisfying to break one of the many keys (instead of the key of a specific computer/session)

Extending dOAE to the multi-user setting

- A natural extension: just include multiple users into the security game

proc initialize **Game dReal**

$J_1, \ldots, J_u \leftarrow 0$; $\mathcal{X}_1, \ldots, \mathcal{X}_u \leftarrow \emptyset$;
$(K_1, \ldots, K_u) \xleftarrow{\$} (\mathcal{K})^u$;

proc Enc.init(i, N)
if $N \notin \mathcal{N}$ **then return** $\perp$
$J_i \leftarrow J_i + 1$; $N_{i,J_i} \leftarrow N$;
$\boldsymbol{A}_{i,J_i} \leftarrow \boldsymbol{M}_{i,J_i} \leftarrow \Lambda$
$S_{i,J_i} \leftarrow \mathcal{E}_{K_i}$.init$(N)$
return J_i

proc Enc.seg(i, j, A, M, b) // $b \in \{0,1\}$
if $j \notin [1..J_i]$ **or** $S_{i,j} = \perp$, **then return** $\perp$
if $(N_{i,j}, \boldsymbol{A}_{i,j}\|A, \boldsymbol{M}_{i,j}\|M', b) \in \mathcal{X}_i$, for some $M' \neq M$, **then return** $\perp$
if $b = 0$, $(C, S_{i,j}) \leftarrow \mathcal{E}_{K_i}$.next$(S_{i,j}, A, M)$
if $b = 1$, $(C, S_{i,j}) \leftarrow \mathcal{E}_{K_i}$.last$(S_{i,j}, A, M)$
$\boldsymbol{A}_{i,j} \leftarrow \boldsymbol{A}_{i,j}\|A$; $\boldsymbol{M}_{i,j} \leftarrow \boldsymbol{M}_{i,j}\|M$;
$\mathcal{X}_i \leftarrow \mathcal{X}_i \cup \{(N_{i,j}, \boldsymbol{A}_{i,j}, \boldsymbol{M}_{i,j}, b)\}$
return C

proc initialize **Game dRand**

$J_1, \ldots, J_u \leftarrow 0$; $\mathcal{X}_1, \ldots, \mathcal{X}_u \leftarrow \emptyset$;
$E_i(x) \leftarrow$ undef for all x $(1 \le i \le u)$

proc Enc.init(i, N)
if $N \notin \mathcal{N}$ **then return** $\perp$
$J_i \leftarrow J_i + 1$; $N_{i,J_i} \leftarrow N$;
$\boldsymbol{A}_{i,J_i} \leftarrow \boldsymbol{M}_{i,J_i} \leftarrow \Lambda$
return J_i

proc Enc.seg$(i, j, A, M, 0)$ // $b \in \{0,1\}$
if $j \notin [1..J_i]$ **or** $N_{i,j} = \perp$, **then return** $\perp$
if $(N_{i,j}, \boldsymbol{A}_{i,j}\|A, \boldsymbol{M}_{i,j}\|M', b) \in \mathcal{X}_i$, for some $M' \neq M$, **then return** $\perp$
$\boldsymbol{A}_{i,j} \leftarrow \boldsymbol{A}_{i,j}\|A$; $\boldsymbol{M}_{i,j} \leftarrow \boldsymbol{M}_{i,j}\|M$
$\mathcal{X}_i \leftarrow \mathcal{X}_i \cup \{(N_{i,j}, \boldsymbol{A}_{i,j}, \boldsymbol{M}_{i,j}, b)\}$
if $E_i(N_{i,j}, \boldsymbol{A}_{i,j}, \boldsymbol{M}_{i,j}, b) =$ undef, **then** $E_i(N_{i,j}, \boldsymbol{A}_{i,j}, \boldsymbol{M}_{i,j}, b) \xleftarrow{\$} \{0,1\}^{|M|+\tau}$
$C \leftarrow E_i(N_{i,j}, \boldsymbol{A}_{i,j}, \boldsymbol{M}_{i,j}, b)$
if $b = 1$, **then** $N_{i,j} \leftarrow \perp$
return C

Fig. 1: Multi-user dOAE privacy experiments.

proc initialize **Game dForge**

$J_1, \ldots, J_u \leftarrow 0$; $\mathcal{X}_1, \ldots, \mathcal{X}_u \leftarrow \emptyset$; $(K_1, \ldots, K_u) \xleftarrow{\$} (\mathcal{K})^u$;

proc Enc.init(i, N)
if $N \notin \mathcal{N}$ **then return** $\perp$
$J_i \leftarrow J_i + 1$; $N_{i,J_i} \leftarrow N$; $\boldsymbol{A}_{i,J_i} \leftarrow \boldsymbol{M}_{i,J_i} \leftarrow \boldsymbol{C}_{i,J_i} \leftarrow \Lambda$; $S_{i,J_i} \leftarrow \mathcal{E}_{K_i}$.init$(N)$
return J_i

proc Enc.seg(i, j, A, M, b) // $b \in \{0,1\}$
if $j \notin [1..J_i]$ **or** $S_{i,j} = \perp$, **then return** $\perp$
if $(N_{i,j}, \boldsymbol{A}_{i,j}\|A, \boldsymbol{M}_{i,j}\|M', b) \in \mathcal{X}_i$, for some $M' \neq M$, **then return** $\perp$
if $b = 0$, **then** $(C, S_{i,j}) \leftarrow \mathcal{E}_{K_i}$.next$(S_{i,j}, A, M)$, **else** $(C, S_{i,j}) \leftarrow \mathcal{E}_{K_i}$.last$(S_{i,j}, A, M)$
$\boldsymbol{A}_{i,j} \leftarrow \boldsymbol{A}_{i,j}\|A$; $\boldsymbol{M}_{i,j} \leftarrow \boldsymbol{M}_{i,j}\|M$; $\boldsymbol{C}_{i,j} \leftarrow \boldsymbol{C}_{i,j}\|C$;
$\mathcal{X}_i \leftarrow \mathcal{X}_i \cup \{(N_{i,j}, \boldsymbol{A}_{i,j}, \boldsymbol{M}_{i,j}, b)\}$; $\mathcal{Y}_i \leftarrow \mathcal{Y}_i \cup \{(N_{i,j}, \boldsymbol{A}_{i,j}, \boldsymbol{C}_{i,j}, b)\}$
return C

proc finalize $(i, N, \boldsymbol{A}, \boldsymbol{C}, b)$ // $b \in \{0,1\}$
if $(N, \boldsymbol{A}, \boldsymbol{C}, b) \in \mathcal{Y}_i$ **or** $|\boldsymbol{A}| \neq |\boldsymbol{C}|$ **or** $|\boldsymbol{A}| = 0$, **then return** false
$S \leftarrow \mathcal{D}_{K_i}$.init$(N)$; $m \leftarrow |\boldsymbol{C}|$
$j \leftarrow 1$ **to** $m - b$ **do**
$(M, S) \leftarrow \mathcal{D}_{K_i}$.next$(S, \boldsymbol{A}[j], \boldsymbol{C}[j])$
if $M = \perp$ **then return** false
if $\mathcal{D}_{K_i}$.last$(S, \boldsymbol{A}[m], \boldsymbol{C}[m]) = \perp$ **and** $b = 1$, **then return** false
return true

Fig. 2: Multi-user dOAE integrity experiment.

■ Multi-user dOAE security of SpookChain

Assuming that the number of involved users $u \leq 2^{n_p}$, $n_p \leq n$, $n \geq 5$, and $2\sigma + 2\sigma_d + q_\pi \leq Min\{2^n/4, 2^{r+c}/2\}$. Then for any adversary it holds

- $Adv^{doae.priv}_{SpookChain} \leq \frac{3u}{2^{n_p}} + \frac{4n(2\sigma+q_\pi)+nq_{IC}+n^2q_e}{2^n} + \frac{17(2\sigma+q_\pi)^2}{2^c}$,
- $Adv^{doae.auth}_{SpookChain} \leq \frac{3u}{2^{n_p}} + \frac{4n(2\sigma+2\sigma_d+q_\pi)+nq_{IC}+2}{2^n} + \frac{16(2\sigma+2\sigma_d+q_\pi)^2}{2^c}$.

■ Multi-user dOAE security of SpookChain

Assuming that the number of involved users $u \leq 2^{n_p}$, $n_p \leq n$, $n \geq 5$, and $2\sigma + 2\sigma_d + q_\pi \leq Min\{2^n/4, 2^{r+c}/2\}$. Then for any adversary it holds

- $Adv^{doae.priv}_{SpookChain} \leq \frac{3u}{2^{n_p}} + \frac{4n(2\sigma+q_\pi)+nq_{IC}+n^2q_e}{2^n} + \frac{17(2\sigma+q_\pi)^2}{2^c}$,
- $Adv^{doae.auth}_{SpookChain} \leq \frac{3u}{2^{n_p}} + \frac{4n(2\sigma+2\sigma_d+q_\pi)+nq_{IC}+2}{2^n} + \frac{16(2\sigma+2\sigma_d+q_\pi)^2}{2^c}$.

- The term $\sigma^2/2^c$ is typical in sponge-based modes.

■ Multi-user dOAE security of SpookChain

Assuming that the number of involved users $u \leq 2^{n_p}$, $n_p \leq n$, $n \geq 5$, and $2\sigma + 2\sigma_d + q_\pi \leq Min\{2^n/4, 2^{r+c}/2\}$. Then for any adversary it holds

- $Adv^{doae.priv}_{SpookChain} \leq \frac{3u}{2^{n_p}} + \frac{4n(2\sigma+q_\pi)+nq_{IC}+n^2q_e}{2^n} + \frac{17(2\sigma+q_\pi)^2}{2^c}$,
- $Adv^{doae.auth}_{SpookChain} \leq \frac{3u}{2^{n_p}} + \frac{4n(2\sigma+2\sigma_d+q_\pi)+nq_{IC}+2}{2^n} + \frac{16(2\sigma+2\sigma_d+q_\pi)^2}{2^c}$.

- The term $\sigma^2/2^c$ is typical in sponge-based modes.
- The birthday term $\sigma^2/2^n$ often appears in modes built upon n-bit (tweakable) blockciphers, though we have overcame.

How to prove?

- The proof is based on a tree-based argument originally established for keyed sponges
- Nodes in the trees are marked by the value of the c-bit capacity halves
- Forgery means the adversary succeeds in adding a new leaf via decryption queries
- We show that collisions on these values appear with probability $\sigma^2/2^c$
- And if such collisions do not occur, there is no method to add new leaves

How to prove?

- The proof is based on a tree-based argument originally established for keyed sponges
- Nodes in the trees are marked by the value of the c-bit capacity halves
- Forgery means the adversary succeeds in adding a new leaf via decryption queries
- We show that collisions on these values appear with probability $\sigma^2/2^c$
- And if such collisions do not occur, there is no method to add new leaves
- See the arguments in the paper

For Cryptographic Engineering

- We did not try to prove (or even formalize) the leakage security of SpookChain.

For Cryptographic Engineering

- We did not try to prove (or even formalize) the leakage security of SpookChain.
- But since the mode stems from Spook or TETSponge, we expect it to ensure side-channel security with much lighter side-channel protections.
  - Should be friendly to cryptographic engineering

山东大学
SHANDONG UNIVERSITY

Thank you!
Questions or comments?

References

- Bellare, M., Boldyreva, A., Knudsen, L.R., Namprempre, C.: Online ciphers and the Hash-CBC construction. CRYPTO 2001.
- Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. SAC 2011
- Fleischmann, E., Forler, C., Lucks, S.: McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. FSE 2012.
- Guo, C., Pereira, O., Peters, T., Standaert, F.: Towards Lightweight Side-Channel Security and the Leakage-Resilience of the Duplex Sponge. IACR Cryptology ePrint Archive 2019, 133 (2019)
- Hoang, V.T., Reyhanitabar, R., Rogaway, P., Vizar, D.: Online Authenticated-Encryption and its Nonce-Reuse Misuse-Resistance. CRYPTO 2015.